



GP-2764

#9
3/19/02

Docket No. 500.37238X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): KOBAYASHI, et al.

Serial No.: 09/317,434

Filed: May 24, 1999

Title: METHOD AND SYSTEM FOR MANAGING DATABASES
HAVING A CAPABILITY OF PASSING DATA, AND
MEDIUM RELEVANT THERETO

Group:

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

June 15, 1999

Sir:

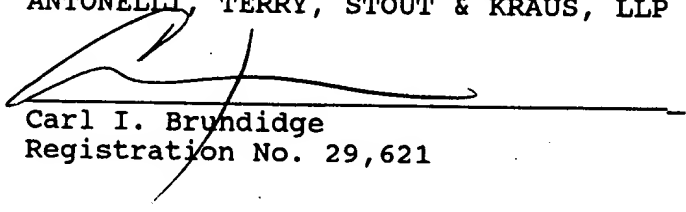
Under the provisions of 35 U.S.C. 119 and 37 CFR 1.55, the
applicant(s) hereby claim(s) the right of priority based on:

Japanese Patent Application No. 10-147262
Filed: May 28, 1998

A Certified Copy of said Japanese Patent Application is
attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP


Carl I. Brundidge
Registration No. 29,621

CIB/ssr
Attachments



日 本 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

1998年 5月28日

出 願 番 号
Application Number:

平成10年特許願第147262号

出 願 人
Applicant(s):

株式会社日立製作所

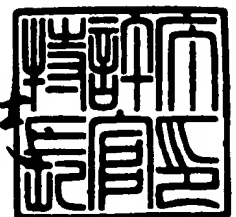
RECEIVED
99 JUN 16 AM 9:48
GROUP 2700

CERTIFIED COPY OF
PRIORITY DOCUMENT

1999年 5月21日

特許庁長官
Commissioner,
Patent Office

山 佐 建 志



出証番号 出証特平11-3031539

【書類名】 特許願

【整理番号】 K98001861

【提出日】 平成10年 5月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00

【発明の名称】 データベース処理方法、データベース処理システム及び媒体

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県横浜市都筑区加賀原二丁目2番 株式会社日立製作所 システム開発本部内

【氏名】 小林 拳

【発明者】

【住所又は居所】 神奈川県横浜市都筑区加賀原二丁目2番 株式会社日立製作所 システム開発本部内

【氏名】 山本 洋一

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア開発本部内

【氏名】 亀城 嘉人

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100083552

【弁理士】

【氏名又は名称】 秋田 収喜

【電話番号】 03-3893-6221

【手数料の表示】

【予納台帳番号】 014579

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データベース処理方法、データベース処理システム及び媒体

【特許請求の範囲】

【請求項 1】 長大データを取り扱うクライアントーサーバ構成のデータベースシステムにおいて、サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する第 1 の過程と、該第 1 の過程で長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する第 2 の過程とを含むことを特徴とするデータベース処理方法。

【請求項 2】 請求項 1 に記載のデータベース処理方法において、前記長大データを出力するファイルを識別するファイル識別情報を前記データベースサーバが作成する第 3 の過程と、前記ファイル識別情報を前記データベースサーバから前記ユーザアプリケーションに通知する第 4 の過程と、前記通知により取得した前記ファイル識別情報を用いて前記ユーザアプリケーションがファイルを参照して長大データを取得する第 5 の過程とを含むことを特徴とするデータベース処理方法。

【請求項 3】 長大データを取り扱うクライアントーサーバ構成のデータベースシステムにおいて、サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する手段と、該手段により長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する手段とを備えることを特徴とするデータベース処理システム。

【請求項 4】 請求項 3 に記載のデータベース処理システムにおいて、前記長大データを出力するファイルを識別するファイル識別情報を前記データベースサーバが作成する手段と、前記ファイル識別情報を前記データベースサーバから前記ユーザアプリケーションに通知する手段と、前記通知により取得した前記ファイル識別情報を用いて前記ユーザアプリケーションがファイルを参照して長大データを取得する手段とを備えることを特徴とするデータベース処理システム。

【請求項5】 クライアントーサーバ構成のデータベースシステムにおけるプログラム及びデータを記録したコンピュータ読み取り可能な記憶媒体であって、前記サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する手順と、該手順により長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する手順とを備えるデータベース処理システムのプログラム及びデータを記録した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データベース処理方法、データベース処理システム及び媒体に関し、特に、クライアントーサーバ構成のデータベースシステムにおいて長大データを処理する方法に関する。

【0002】

【従来の技術】

動画・画像データや音声データなどひとかたまりのデータが数ギガバイトにも及ぶような長大なデータをデータベースで管理することに関して、SQL3 (Database Language SQL, ISO Working Draft, July 1996) で提案されているBLOB(Binary Large Object)型を用いる方法がある。

【0003】

長大データをデータベースシステムで扱う場合、ユーザアプリケーションの動作するプログラムで膨大なメモリ領域が必要になることが問題となる。この問題に対して、BLOBデータをファイルに保持することにより、アプリケーションプログラムで長大データを保持するためのメモリバッファを使用せずにBLOBデータを扱う方法として、データベース管理システムDB2("USING THE NEW DB2 IBM's Object-Relational Database System", Don Chamberlin, Morgan Kaufmann Publishers, Inc., 1996)のファイルリファレンス(file reference)の技術がある。

【0004】

一方、リレーショナルデータベースでは、膨大なデータに対しても高速に処理

することを可能にする並列データベースの技術がある。並列データベースにおいて問い合わせの内容やデータの状態に応じて最適な並列処理を選択する機能をもつシステム（特開平6-214843）がある。

【0005】

また、オブジェクト・リレーショナルデータベース（「オブジェクトリレーショナルDBMSs」 Michael Stonebraker著、大田佳伸訳、インターナショナル・トムソン・パブリッシング・ジャパン、1996年8月）では、SQL3のルーチンの機能を汎用のプログラミング言語で記述して作成される実行可能なコードを用いて実現し、そのルーチンを用いたSQLによって画像などのマルチメディアデータを操作できるようにする技術がある。

【0006】

【発明が解決しようとする課題】

前記の従来の公知技術においては次のような問題点がある。

【0007】

（1）前記のファイルリファレンスの方法では、アプリケーションプログラム（AP）の動作するクライアントのノード（コンピュータマシン。中央処理装置を基本とするコンピュータを構成する単位）上にファイルを置き、そのファイルにBLOBデータを保持することになっている。（このようにデータベースサーバの外部で扱うファイルを、以降、外部ファイルと呼ぶ。）サーバに保持しているBLOBデータをファイルに出力するようAPから要求がある場合、以下のように処理が行われる。

- （a）サーバノードでデータベースサーバがBLOBデータを読み込む。
- （b）サーバからクライアントにデータを転送する。
- （c）クライアントでデータを外部ファイルに書き出す。

この場合、クライアントーサーバ間でネットワーク通信またはプロセス間通信で長大なデータを転送するため、処理に時間がかかるという問題があった。

【0008】

（2）前記のファイルリファレンスの方法では、APが指定した外部ファイルにデータベースサーバがBLOBデータを出力することになっている。

そのため、例えば、データベース問い合わせの結果として得られる複数件のBLOBデータをそれぞれ個別の外部ファイルに出力するには、以下のような煩雑な処理を行なう必要がある。

(a) BLOBデータが外部ファイルに出力されるときに上書きされるように指定しておき、結果を一件出力するたびに、外部ファイルの名称を別の名称に変更するか、外部ファイルの内容を別のファイルにコピーする。

(b) BLOBデータが外部ファイルに出力されるときに追加されるように指定しておき、後に外部ファイルからそれぞれのBLOBデータを分割して、それぞれ別のファイルに出力する。なお、BLOBデータのサイズが不明の場合は、結果を一件出力するたびに、追加されたBLOBデータのサイズを保持しておいて、後にそのサイズを参照しながらBLOBデータを分割する。

したがって、このような処理を行なうために、APのソースコードの記述が煩雑になるという問題があった。

また、長大なBLOBデータのコピーなどを行なう場合は、メモリが大きくなるという問題があった。

【0009】

(3) 並列データベースシステムにおいて、前記のファイルリファレンスの方法を適用する場合、データベースサーバが複数の並列データベース処理によって複数のBLOBデータに対して並列にアクセスするのにもかかわらず、外部ファイルへのBLOBデータの書き出しはAPのある1つのクライアントで一件ずつ行われることになる。

そのため、クライアントに負荷が集中してボトルネックとなり、性能上問題となる。

【0010】

本発明の目的は、前記(1)から(3)の問題点を解決し、データベースシステムで外部ファイルを介して長大なデータを扱う処理を高速化することが可能な技術を提供することにある。

本発明の他の目的は、ユーザアプリケーションで外部ファイルを介して長大なデータを扱うためのソースコードの記述を簡易にすることが可能な技術を提供す

ることにある。

本発明の他の目的は、並列データベースシステムの場合に、複数の外部ファイルの出力を高速化することが可能な技術を提供することにある。

本発明の前記ならびにその他の目的及び新規な特徴は、本明細書の記述及び添付図面によって明らかにする。

【0011】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、以下のとおりである。

【0012】

(1) 長大データを取り扱うクライアントーサーバ構成のデータベースシステムにおいて、サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力し、この長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得するデータベース処理方法である。

【0013】

(2) 前記長大データを出力するファイルを識別するファイル識別情報を前記データベースサーバが作成し、前記ファイル識別情報を前記データベースサーバから前記ユーザアプリケーションに通知し、前記通知により取得した前記ファイル識別情報を用いて前記ユーザアプリケーションがファイルを参照して長大データを取得するデータベース処理方法である。

【0014】

(3) 前記ユーザアプリケーションがデータベースに定義された関数の実行を要求し、前記ユーザアプリケーションからの要求に応じて前記データベースサーバが前記関数を実行し、前記関数が前記長大データを出力するファイルのファイル識別情報を作成し、前記関数が前記長大データをファイルに出力し、前記関数が前記ファイル識別情報を前記データベースサーバに通知するデータベース処理方法である。

【0015】

(4) 並列データベース構成で並列にデータベース処理を行なう複数のプロセスが、それぞれ並列に前記長大データをファイルに出力するデータベース処理方法である。

【0016】

(5) 前記データベースサーバにより長大データが出力されたファイルを前記データベースサーバが動作するノードと同一のノードで前記ユーザアプリケーションが参照して長大データを取得するデータベース処理方法である。

【0017】

(6) 長大データを取り扱うクライアントーサーバ構成のデータベースシステムにおいて、サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する手段と、該手段により長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する手段とを備えるデータベース処理システムである。

【0018】

(7) 前記長大データを出力するファイルを識別するファイル識別情報を前記データベースサーバが作成する手段と、前記ファイル識別情報を前記データベースサーバから前記ユーザアプリケーションに通知する手段と、前記通知により取得した前記ファイル識別情報を用いて前記ユーザアプリケーションがファイルを参照して長大データを取得する手段とを備えるデータベース処理システムである。

【0019】

(8) 前記ユーザアプリケーションがデータベースに定義された関数の実行を要求する手段と、前記ユーザアプリケーションからの要求に応じて前記データベースサーバが前記関数を実行する手段と、前記関数が前記長大データを出力するファイルのファイル識別情報を作成する手段と、前記関数が前記長大データをファイルに出力する手段と、前記関数が前記ファイル識別情報を前記データベースサーバに通知する手段とを備えるデータベース処理システムである。

【0020】

(9) 並列データベース構成で並列にデータベース処理を行なう複数のプロセスが、それぞれ並列に前記長大データをファイルに出力する手段を備えるデータベース処理システムである。

【0021】

(10) 前記データベースサーバにより長大データが出力されたファイルを前記データベースサーバが動作するノードと同一のノードで前記ユーザアプリケーションが参照して長大データを取得する手段を備えるデータベース処理システムである。

【0022】

(11) クライアントサーバ構成のデータベースシステムにおけるプログラム及びデータを記録したコンピュータ読み取り可能な記憶媒体であって、前記サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する手順と該手順により長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する手順とを備えるデータベース処理システムのプログラム及びデータを記録した記憶媒体である。

【0023】

【発明の実施の形態】

以下、本発明の実施形態（実施例）を図面を用いて詳細に説明する。

【0024】

図1は本発明の実施形態の並列データベースシステムにおいて、長大データを外部ファイルを介して扱う処理の概要を示す概念図である。

図1に示すように、本実施形態のデータベースシステム101は、アプリケーションプログラム(AP)112が動作するクライアントとなるノード111と、AP112からの問い合わせ要求を受け付けるデータベースサーバ102の機能の一部であるフロントエンドサーバ(FES)122の動作するノード121と、データベースシステム101における定義情報や格納構造情報などを含むディクショナリ情報133を管理するデータベースサーバ102の機能の一部であるデ

ィクシヨナリサーバ (DS) 132が動作するノード131と、データベースシステム101におけるデータベース処理を並列して実行するデータベースサーバ102の機能の一部である複数のバックエンドサーバ (BES) 142-1、142-2、…、142-nが動作する複数のノード141-1、141-2、…、141-nと、外部ファイル152-1、152-2、…、152-nを保持するノード151-1、151-2、…151-nは、ネットワーク103によって相互結合されている。

【0025】

BES142-1、142-2、…、142-nは、AP112の操作対象となるBLOBデータ145-1、145-2、…、145-nをデータストア144-1、144-2、…、144-nに保持している。

【0026】

ノード151-1、151-2、…、151-nでは、AP112及びデータベースサーバ102が共通にアクセスする共用のファイルシステム領域に、外部ファイル152-1、152-2、…、152-nを保持する。

【0027】

AP112は問い合わせ要求をデータベースサーバ102のFES122に伝える(161)。この問い合わせ要求には、BLOBデータ145-1、145-2、…、145-nを外部ファイル152-1、152-2、…、152-nに出力する関数の実行要求113が含まれている。

【0028】

FES122は、AP112からの問い合わせ要求の解析を行なう。問い合わせ要求の解析では、DS132のディクシヨナリ情報133を参照して(162)、データストア144-1、144-2、…、144-nの格納構造情報や関数実行に関する情報を取得する。次に、FES122は、問い合わせ要求を解析した情報をもとに、データベース処理の実行手順コード123を作成し、実行手順コード123としてBES142-1、142-2、…、142-nへ送出し、データベース処理を行なうように要求する(163-1、163-2、…、163-n)。実行手順コード123には、BLOBデータ145-1、145-2、…、145-n

を外部ファイル 152-1、152-2、…、152-n に書き出す機能を持つ関数の実行指示が含まれる。

【0029】

BES142-1、142-2、…、142-n は実行手順コード 123 に従い、BLOB データ 145-1、145-2、…、145-n をデータストア 144-1、144-2、…、144-n から読み込み、BLOB データを外部ファイルを書き出す関数の関数実装コード 143-1、143-2、…、143-n を実行し、外部ファイル 152-1、152-2、…、152-n を作成する (164-1、164-2、…、164-n)。

【0030】

関数実装コード 143-1、143-2、…、143-n を実行した結果、BES142-1、142-2、…、142-n は外部ファイルの名称を実行結果として受け取る。外部ファイルを作成したのち、BES142-1、142-2、…、142-n は実行結果を FES122 に返信する。

【0031】

FES122 は BES142-1、142-2、…、142-n からの実行結果を編集して、AP112 に処理結果として返信する。

【0032】

AP112 は、返信された処理結果に含まれる外部ファイル名を取得し、外部ファイル 152-1、152-2、…、152-n の内容の BLOB データを参照する (165-1、165-2、…、165-n)。

【0033】

これにより、クライアント-サーバ間で長大データを転送することなく、データベースサーバが複数の外部ファイルを並列に書き出すので、高速に処理することができる。また、複数の外部ファイルのそれぞれのファイル名がデータベースサーバで作成されるので、ユーザアプリケーションでのソースコードの記述を簡易にすることができる。

【0034】

図 2 は図 1 に示す本発明の実施形態 (実施例) におけるハードウェアの構成を

示す概略図である。

図2に示すように、データベースシステム101では、それぞれが処理装置の単位である複数のノード111、121、131、141-1、141-2、…、141-n、151-1、151-1、…、151-nが、LAN(Local Area Network)などのネットワーク103で接続されており、それぞれのノードがネットワーク103を介して通信を行なうことができる。

【0035】

前記ノード111、121、131、141-1、141-2、…、141-n、151-1、151-1、…、151-nは、それぞれ一般的なコンピュータ構成をもち、中央演算装置(CPU)211a、211b、211c、211d、211e、OS212a、212b、212c、212d、212e、ネットワーク103に接続される通信コントローラ213a、213b、213c、213d、213e、主記憶装置(メモリ)214a、214b、214c、214d、214e、I/Oコントローラ215a、215b、215c、215d、215e及びこれらを接続するシステムバス216a、216b、216c、216d、216eなどを含むデータ処理装置201a、201b、201c、201d、201eと、I/Oコントローラ215a、215b、215c、215d、215eに接続されるキーボードやマウス及びディスプレイなどのようなデータ入出力装置202a、202b、202c、202d、202eと、ディスクデバイスのようなデータ記憶装置203a、203b、203c、203d、203eなどから構成される。

【0036】

図1及び図3に示すプログラムの機能は、OS212a、212b、212c、212d、212eの管理のもとで、CPU211a、211b、211c、211d、211eがメモリ214a、214b、214c、214d、214eに格納されたプログラムを実行することにより実現される。

【0037】

図3は図1に示す本発明の実施形態におけるデータベースシステムの機能構成を示す概略図である。

データベースシステム 101 は、AP 112 の動作するクライアントのノード 111 と、AP 112 からの問い合わせ要求を受け付ける FES 122 の動作をするノード 121 と、ディクショナリ情報 133 を管理するディクショナリサーバ (DS) 132 が動作するノード 131 と、データベース処理を並列に実行する複数の BES 12-1、142-2、…、142-n が動作する複数のノード 141-1、141-2、…、141-n と、外部ファイル 152-1、152-2、…、152-n を保持するノード 151-1、151-2、…、151-n から構成される。

【0038】

これらのノードは、ネットワーク 103 によって相互結合されている。AP 112 は、ユーザの要求に応じたデータ処理を行なうコードを含むアプリケーション部 114 と、データベースサーバとの通信を管理するクライアント通信部 115 からなる。アプリケーション部 114 は、BLOB データを外部ファイルに出力する機能をもつ関数の実行を要求するコード 113 を含む。

【0039】

FES 122 は、クライアントからの要求の受信及びクライアントへのデータベース処理結果の返信などを管理する FES 通信部 124 と、クライアントからの要求を解析し、データベース処理の実行手順を示す実行手順コード 123 を生成する要求解析部 125 からなる。

【0040】

BES 142-1、142-2、…、142-n は、FES 122 からの要求の受信及び FES 122 へのデータベース処理結果の返信などを管理する BES 通信部 146 と、実行手順コード 123 に含まれる指示に従ってデータベース処理を行なう処理実行部 147 と、データを保持するデータストア 144-1、144-2、…、144-n へのアクセスを管理するデータアクセス管理部 148-1、148-2、…、148-n からなる。

【0041】

以降では、アプリケーションの例を用いて、本発明を適用したデータベース処理を詳細に説明する。

以下では、データベースシステム 101 において、従業員に関する情報を管理する例を用いる。AP 112 は、設計部署に所属するすべての従業員の名前と写真を問い合わせ、結果を一覧表示するプログラムであるとする。

従業員の情報は、表 employee を用いて表わす。表 employee は、従業員番号を表わすための列 empno、従業員の名前を表わすための列 name、従業員の所属部署を表わすための列 dept、従業員に関する写真データを表わすための列 photo からなる。

【0042】

図 4 に表 employee を定義する SQL 定義文 401 を示す。以下、各行の意味を示す。

402 : employee という名称の表の定義で、以下の列からなる。

403 : INTEGER 型の列で、列名称が empno

404 : VARCHAR 型 (30 バイト以内) の列で、列名称が name

405 : VARCHAR 型 (30 バイト以内) の列で、列名称が dept

406 : BLOB 型 (10 メガバイト以内) の列で、列名称が photo

定義文 401 をもとに一般的なデータベース処理を行なうことによって、ディクショナリ情報 133 に表の定義を登録する。

この登録により、データベースサーバ 102 は、のちにディクショナリ情報 133 を参照し、表における列の構成の情報や、表の定義に従って格納されるデータにアクセスするための情報などが取得できる。

【0043】

図 5 は前記 401 の定義に基づいて従業員に関するデータを格納した表の概要図である。

行データを格納する表 501 は、列 empno 511、列 name 512、列 dept 513、及び列 photo 514 からなる。

【0044】

行データの意味は以下に示す通りである。

521 : 列 empno, name, dept の値がそれぞれ、INTEGER 型の整数 1789, VARCHAR 型の文字列 George, VARCHAR 型の文字列 design であり、列 photo の値は、BLOB デー

タの識別情報(blob1)である。

このデータ521は、従業員番号が1789、名前がGeorge、所属部署が設計部、写真データが(blob1)で識別されるBLOBデータであることを表わしている。

なお、列値のBLOBデータに関しては、BLOBデータの識別情報を列値に保持しておき、BLOBデータの実体はデータストア中の別の領域に保持する一般的な方法によって格納する。

以下、同様の行データについて簡単に意味を示す。

522：列empno, name, deptの値がそれぞれ、INTEGER型の整値1797, VARCHAR型の文字列John, VARCHAR型の文字列accountであり、列photoの値は、BLOBデータの識別情報(blob2)である。

523：列empno, name, deptの値がそれぞれ、INTEGER型の整値1801, VARCHAR型の文字列Thomas, VARCHAR型の文字列designであり、列photoの値は、BLOBデータの識別情報(blob3)である。

524：列empno, name, deptの値がそれぞれ、INTEGER型の整値1809, VARCHAR型の文字列James, VARCHAR型の文字列generalであり、列photoの値は、BLOBデータの識別情報(blob4)である。

525：列empno, name, deptの値がそれぞれ、INTEGER型の整値1829, VARCHAR型の文字列Andrew, VARCHAR型の文字列accountであり、列photoの値は、BLOBデータの識別情報(blob5)である。

526：列empno, name, deptの値がそれぞれ、INTEGER型の整値1837, VARCHAR型の文字列Martin, VARCHAR型の文字列planningであり、列photoの値は、BLOBデータの識別情報(blob6)である。

【0045】

次に、BLOBデータを外部ファイルに書き出す機能を提供する関数fileout()について説明する。

関数fileout()のインタフェース仕様は、入力(関数の引数)が、BLOB型のデータであり、出力(関数の戻り値)が、VARCHAR型(255バイト以内)の文字列である。

この関数fileout()の機能は、引数で入力されたBLOBデータの内容を外部ファ

イルに書き出すことであり、BLOBデータの内容を書き出す外部ファイルの名称は、本関数自体で作成する。そして、BLOBデータの内容を書き出した外部ファイルの名称を関数の戻り値として返す。

【0046】

図3のノード151-1、151-2、…、151-nで共用のファイルシステム領域に外部ファイルを作成する。

【0047】

本関数fileout()の処理内容については、図14で説明する。

【0048】

関数fileout()は図6のようなSQL定義文で定義する。

関数の定義文601の各行の意味を以下に示す。

602: fileoutという名前の関数の定義であり、BLOB型の値を引数とする。

603: 戻り値がVARCHAR型(255バイト以内)の値である。

604: この関数の機能を実装している実行可能なコード(関数実装コード)をbin/fileoutというファイルに保持している。

605: 関数実装コードのソースコードはC言語で記述されている。

【0049】

定義文601をもとに一般的なデータベース処理を行なうことによって、ディクショナリ情報133(図3)に関数の定義を登録する。

この登録により、データベースサーバ102は、のちにディクショナリ情報133を参照し、関数fileout()の入出力や関数実装コードがbin/fileoutであることなどの関数を実行するための情報を取得することができる。

【0050】

図7は、AP112のアプリケーション部114における処理の概要を示すフローチャートである。

本図では、設計部署に所属するすべての従業員の写真データを一覧表示する処理(701)を示している。

まず、「設計部署に所属するすべての従業員の名前と写真データを取得せよ」という問い合わせ要求を行なう(702)。

次に、問い合わせ要求の結果を一件取り出す（703）。取り出した結果のデータが得られたか判定する（704）。取り出した結果のデータがある場合は、結果のデータを一覧表に設定し（705）、703のステップに戻る。

ステップ704で取り出した結果のデータがない場合、すなわち、すべての検索結果データを取得した場合には、結果データの一覧表を出力して（706）、処理を終了する（707）。

【0051】

図8は図7の処理を行なうアプリケーションを作成するためのソースコードの一部を示す。

アプリケーションのソースコードの一部801の各行の意味を以下に示す。

802：SQL変数の宣言の開始

803：変数photoFilenameの宣言（外部ファイル名を保持する）

804：変数empNameの宣言（従業員名を保持する）

805：SQL変数の宣言の終了

806：行807～809のSQL検索文に対するカーソルcurs1を定義

807：検索の射影項目が、列nameと、列photoを引数とする関数fileout()のコールである

808：検索対象の表がemployee

809：検索の条件は、列deptの値が文字列値design（所属部署が設計部）であること

810：カーソルのオープン要求

811：以下、行815までの処理を繰り返す

812：カーソルcurs1に対するフェッチ要求。結果を変数empName, photoFilenameに保持する。

813：フェッチした結果のデータがない場合は、繰り返し処理を抜ける。

814：関数setImageDataToList()をコールする。引数に従業員名と外部ファイル名を渡す。ここで、関数setImageDataToList()は、図示していないがアプリケーションにリンクされたコードであり、一般的なりスト構造をもつデータに結果データを設定する機能をもつ。

815: 行 811 からの繰り返し処理の範囲の終わりを示す。

816: カーソルのクローズ要求、

817: 関数 `displayImageDataList()` をコールする。

【0052】

ここで、関数 `displayImageDataList()` は、図示していないがアプリケーションにリンクされたコードであり、関数 `setImageDataToList()` で設定された結果データのリストをもとに、結果の一覧の HTML (Hypertext Markup Language) テキストファイルを作成し、HTML ブラウザで一覧を出力する簡単なデータ操作機能をもつ。画像ファイルを読み込んで表示する。

【0053】

ソースコード 801 では、図 4 に定義した表 `employee` 及び図 6 に示した関数 `fileout()` を利用している。

また、データベースサーバが外部ファイル名を返却しており、ソースコードでファイル名を作成したりファイルをコピーする必要のない、簡単な記述になっている。

【0054】

なお、ソースコード 801 を用いて、以下のような一般的な手順で AP112 を作成する。

(1) アプリケーションのソースコードを、一般的にデータベースシステムのアプリケーション開発用のツールに含まれるプリプロセッサを通してホスト言語のソースコードに変換する。

(2) ホスト言語のコンパイラを用いて、変換したソースコードからオブジェクトコードを作成する。このオブジェクトコードが AP112 のアプリケーション部 114 となる。特に、SQL 文で関数をコールする部分 (行 806 ~ 810、811) をもとに、関数実行要求 113 が作成される。

(3) クライアント通信部 115 の機能をもつコードを含むライブラリとリンクすることにより AP を生成する。

【0055】

図 9 は、図 7 及び図 8 で示した AP112 の処理中に、図 11 ~ 図 14 に示す

処理によって得られる検索結果を示す図である。

データベースの検索結果は、ソースコードの行807～809で指定した検索要求文に応じて、設計部署に所属する従業員の名前とその従業員に関する画像データを保持する外部ファイルの名称とを組みとする行を含む表901となる。

【0056】

検索結果の表901は、検索文の射影項目にnameと指定したことに対する列(911)と、検索文の射影項目にfileout(photo)と指定したことに対する列(912)からなる。

各行データの意味は次の通りである。

921: 列911の値がGeorge、列912の値がdbsvexfile1である。

これは、従業員の名前がGeorgeで、その従業員に関する写真データがdbsvexfile1という名称の外部ファイルに保持されていることを示している。

922: 列911の値がThomas、列912の値がdbsvexfile2である。

これは、従業員の名前がThomasで、その従業員に関する写真データがdbsvexfile2という名称の外部ファイルに保持されていることを示している。

【0057】

図10は図9の検索結果をもとにAP112が作成する結果一覧のHTMLテキストである。

HTMLテキスト(1001)の各行の意味を以下に示す。

1002: HTMLテキストの開始、

1003: 文字列George:と、ファイル名dbsvexfile1に保持している画像データを表示して、改行する指示。

1004: 文字列Thomas:と、ファイル名dbsvexfile2に保持している画像データを表示して、改行する指示、

1005: HTMLテキストの終了、

AP112は、このHTMLテキスト1001を、一般的なHTMLブラウザ機能の入力にすることにより、結果一覧(従業員名とその従業員に関する写真)を表示する。HTMLブラウザ機能からファイルdbsvexfile1、dbsvexfile2の画像データが読み込まれて表示される。

【0058】

図11はAP112における問い合わせ要求から結果取得までの処理の概要を示すフローチャートである。

図7のフローチャートのステップ703で問い合わせ要求し、結果を取得する処理(1101)を示す。

まず、AP112がアプリケーション部114に含まれる問い合わせ要求を実行する(1102)。この問い合わせ要求は、クライアント通信部115が提供するデータベース処理実行要求発信の機能をコールする。

次に、クライアント通信部115が、データベースサーバ102のFES122に問い合わせ要求を送信する(1103)。FES122において問い合わせ要求を受信したのちのデータベース処理の詳細については、図12～図14を用いて後述する。

クライアント通信部115は、FES122からの返信待ちの状態になり(1104)、FES122からの返信を受け取ると(1105)、クライアント通信部115は返信されたデータベース処理結果を編集してアプリケーション部114に結果を渡し(1106)、処理を終了する(1107)。

【0059】

図12は図11のステップ1103に応じてFES122において行なわれるデータベース処理(1201)のフローチャートである。

まず、FES122のFES通信部124がクライアント通信部115からの問い合わせ要求を受信する(1202)。

次に、要求解析部125が受信した問い合わせ要求の解析を行なう(1203)。問い合わせ要求の解析では、要求解析部125はDS132に要求してディクショナリ情報133を参照し、APが要求するデータにアクセスするための情報や、関数を実行するための情報などを取得する。

ここでは、表employeeに対する検索要求であることを認識し、図4の定義をもとに図5のように格納された表employeeの行データについて、条件を満たす行データを逐次処理する手順にすることを選定する。

また、問い合わせ中に関数fileout()の実行要求があることを認識し、図6の定

義をもとに関数実行コード143を実行するための情報を取得する。

続いて、要求解析部125は、1203で解析した結果をもとに、BES142-1、142-2、…、142-nでデータベース処理を行なうための実行手順コード123を作成する(1204)。

実行手順コード123には、図5に示したような表employeeに格納される行データへアクセスする手順のコードや、関数実装コード143を実行する手順のコードなどが含まれる。

次に、FES通信部124が実行手順コード123を用いて、データベース処理を実行するように、BES142-1、142-2、…、142-nに処理実行要求を送信する(1205)。

BES142-1、142-2、…、142-nにおけるデータベース処理については、図13で詳細に説明する。

次に、FES通信部124は、BES142-1、142-2、…、142-nからの返信を待つ(1206)。

FES通信部124は、BES142-1、142-2、…、142-nからの返信をがすべてそろったかを判定し(1207)、すべての返信がそろったら、返信に含まれるデータベース処理結果を編集して、問い合わせ要求元であるクライアント通信部115に返信し(1208)、処理を終了する(1209)。

【0060】

図13はBES142-1、142-2、…、142-nにおいて、実行手順コード123に従ってデータベース処理を実行する処理の概要を示すフローチャートである。

図12のステップ1205のデータベース処理実行要求に応じた処理(1301)を示す。

まず、BES通信部146(図3)がFES通信部124からのデータベース処理実行要求を受信する(1302)。

このデータベース処理実行要求には、実行手順コード123が含まれる。

次に、処理実行部147が通信データに含まれる実行手順コード123を解析する(1303)。

ここでは、図5のように格納された表employeeの行データで条件を満たす行データについて、列name、列photoの列値のみを取得し、逐次関数実装コード143を実行すべきことが認識される。以降その解析によって認識した実行手順に従って処理を行なう。

処理実行部147（図3）は、ステップ1303での実行手順コード123の解析結果に従い、検索条件を満たす行データを取得するようデータアクセス管理部148に要求する（1304）。

ここでは、列deptの値が文字列値designであるような行データを要求する。

【0061】

次に行データを取得できたかどうかを判定し（1305）、取得できた場合は、実行手順コード123の解析結果に従い、取得した列photoの値に対して関数実装コード143（bin/fileout）を実行する（1306）。

ここでは、BESが動作しているプログラムが関数実装コードbin/fileoutを動的にリンクして実行する。このとき、列photoの列値であるBLOBデータを扱うためのBLOBロケータを引数として渡す。BLOBロケータは、SQL3で規定されているように、BLOBデータと関連付けられた値とする。BESが動作しているプログラムからは、実際のBLOBデータの内容を直接保持せずに、この値を介してBLOBデータに対する操作を行なうことができるものとする。

関数実装コードの実行の結果、戻り値として、外部ファイル名となるVARCHAR型の値を取得する。

関数実装コードbin/fileoutにおける処理については、図14で説明する。

次に、取得した列値及び関数の戻り値をもとに、データベース処理結果の列データを作成して保持しておき（1307）、1304からの処理を繰り返す。

また、1305で検索条件を満たすデータを取得できなかった場合、すなわち、条件を満たすすべてのデータを取得してしまった場合は、データベース処理結果の列データを編集し（1308）、処理を終了する（1309）。

【0062】

この処理において、列deptの値がdesignである、すなわち、設計部署に所属する521、523の行データ（それぞれ従業員George, Thomasに対応するデータ

）が処理され、図9の結果表の列データ921、922（従業員名と外部ファイル名を組みとする列データ）の元になるデータが作成される。

【0063】

図14は、関数fileout()の実装コードであるbin/fileoutの処理の概要を示すフローチャートである。

本図では、図13のステップ1306において実行されるbin/fileoutの処理（1401）を示す。

まず、図3のノード151-1、151-2、…、151-n上の共用のファイルシステムの領域に、外部ファイル152-1、152-2、…、152-nを作成し、ファイルをオープンする（1402）。

ここで、Cライブラリの関数tempnam()の機能を用いて外部ファイルの名称を作成する。これにより、外部ファイルがこの名称で一意に識別されるようにする。

次に、引数として入力されたBLOBデータのロケータを用いて、メモリ上のバッファへBLOBデータを部分的に読み込み、1402でオープンしたファイルにデータを書き出す（1403）。

次に、BLOBデータをすべて書き出したかどうかを判定する（1404）。

まだすべて書き出していない場合は、1403に戻り、これまでに読み込んだデータの続きを読み出し、外部ファイルに書き出す。

BLOBデータをすべて外部ファイルに書き出した場合は、外部ファイルをクローズし（1405）、外部ファイル名を戻り値として（1406）、処理を終了する（1407）。

【0064】

ここでは、外部ファイルを識別する名称がデータベースサーバによって作成されることになり、前述したようにアプリケーションのソースコードの記述が簡単になる。また、戻り値の外部ファイル名称が処理結果としてAP112に渡され、AP112においても共用のファイルシステム領域にある外部ファイル152-1、152-2、…、152-nを参照することができる。

【0065】

また、並列データベース処理では、BES142-1、142-2、…、142

—nのそれぞれが複数の外部ファイルに並列にBLOBデータを出力するので、前述の公知技術でのクライアントのみで出力する場合に比べて高速に処理することができる。

【0066】

次に、本発明の他の実施例を、図15を用いて説明する。

図15は、1つのノードで構成したデータベースシステムの概要を示す概念図である。

データベースシステム101-aは、1つのノード1501からなり、機能構成は図1および図3に示した構成と同じである。図15には主要な機能のみを示す。

【0067】

この構成において、上記のAP112を実行した場合と同等の処理を行なう。

すなわち、AP112-aがデータベースサーバ102-aに問い合わせ要求を行ない(1511)、データベースサーバ102-aが実装関数コード143-aを実行し、データストア144-aに保持しているBLOBデータ145-aを読み出して(1512)、外部ファイル152-aに出力する(1513)。そして、AP112-aが外部ファイル152-aを参照してBLOBデータ145-aを取得する(1514)。

【0068】

この処理では、AP112-aがBLOBデータ145-aを取得するまでのデータの流れは以下の通りである。

- (1) データベースサーバ102-aがBLOBデータ145-aを読み込む。
- (2) データベースサーバ102-aがBLOBデータ145-aを外部ファイル152-aに出力する。
- (3) AP112-aが外部ファイル152-aからBLOBデータ145-aを読み込む。

【0069】

この間は、BLOBデータをネットワーク通信またはプロセス間通信で転送することはないので、前述した公知技術で必ずデータ転送がある処理と比べて、高速に

処理することができる。

【0070】

なお、上述したフローチャートの処理は、データ処理装置でプログラムを実行することによって実現できる。また、そのプログラムは、ハードディスク装置、フロッピーディスクなどのコンピュータで読み書きができる記憶媒体に格納することができ、ネットワークを通してプログラムにアクセスすることができる。

【0071】

【発明の効果】

クライアント-サーバ構成のデータベースシステムで外部ファイルを介して長大データを扱う場合に、データベースサーバが長大データを外部ファイルに出力し、ユーザアプリケーションが外部ファイルを参照して（外部ファイルを共用して）長大データを取得する手段をもつため、クライアント-サーバ間で長大データを転送することなく、高速に処理することが可能となる。

【0072】

また、データベースサーバが外部ファイルの名称を作成する手段をもつため、アプリケーションで複数の外部ファイルを利用する場合に逐一名称を指定する必要がなく、アプリケーションの記述を簡易することが可能となる。

【0073】

また、並列データベース構成でデータベースサーバが複数のBLOBデータを並列に外部ファイルに出力する手段をもつため、高速に外部ファイルにBLOBデータを出力することが可能となる。

【図面の簡単な説明】

【図1】

本発明の実施形態1の長大データを外部ファイルを介して扱う処理の概要を示す概念図。

【図2】

図1に示す本発明の実施形態におけるハードウェアの構成を示す概略図。

【図3】

図1に示す本発明の実施形態におけるデータベースシステムの機能構成を示す

概略図。

【図 4】

表employeeを定義するSQL定義文を示す図。

【図 5】

表employeeに格納する行データを示す図。

【図 6】

関数fileout()を定義するSQL定義文を示す図。

【図 7】

アプリケーション部における処理のフローチャート。

【図 8】

アプリケーションのソースコードの記述の一部を示す図。

【図 9】

アプリケーションにおける検索結果の表を示す図。

【図 10】

アプリケーションが作成する検索結果一覧を示す図。

【図 11】

APにおける問い合わせ要求から結果取得までの処理のフローチャート。

【図 12】

データベースサーバのFESにおけるデータベース処理のフローチャート。

【図 13】

データベースサーバのBESにおけるデータベース処理のフローチャート。

【図 14】

fileout実装関数の処理のフローチャート。

【図 15】

1つのノードで構成したデータベースシステムの概要を示す概念図である。

【符号の説明】

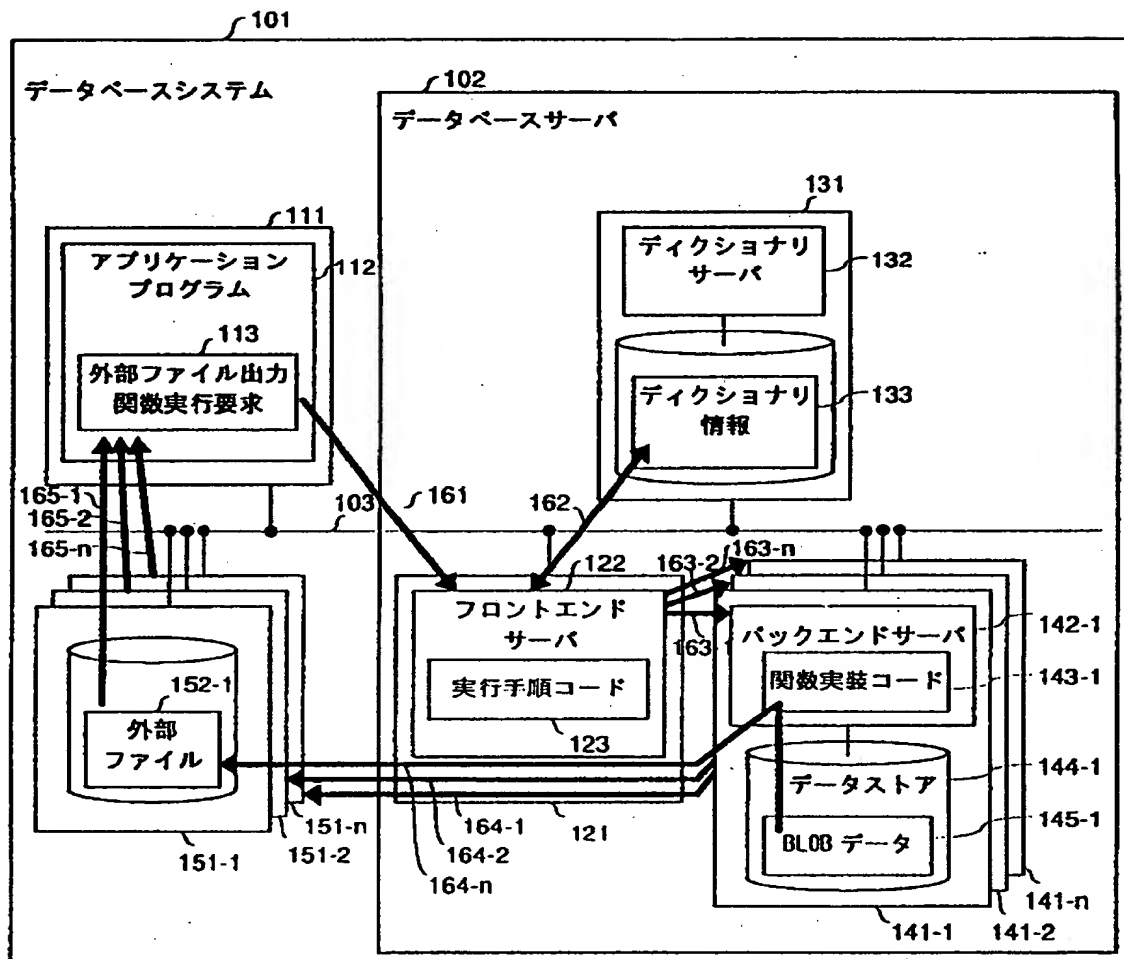
102…データベースサーバ、103…ネットワーク、113…外部ファイル出力関数実行要求、122…データベース処理要求を受け付けるフロントエンドサーバ(FES)、132…ディクショナリ情報を管理するディクショナリサーバ(DS)

、142…データベース処理を実行するバックエンドサーバ(BES)、143…外部ファイル出力の機能を持つ関数実装コード、152…外部ファイル。

【書類名】 図面

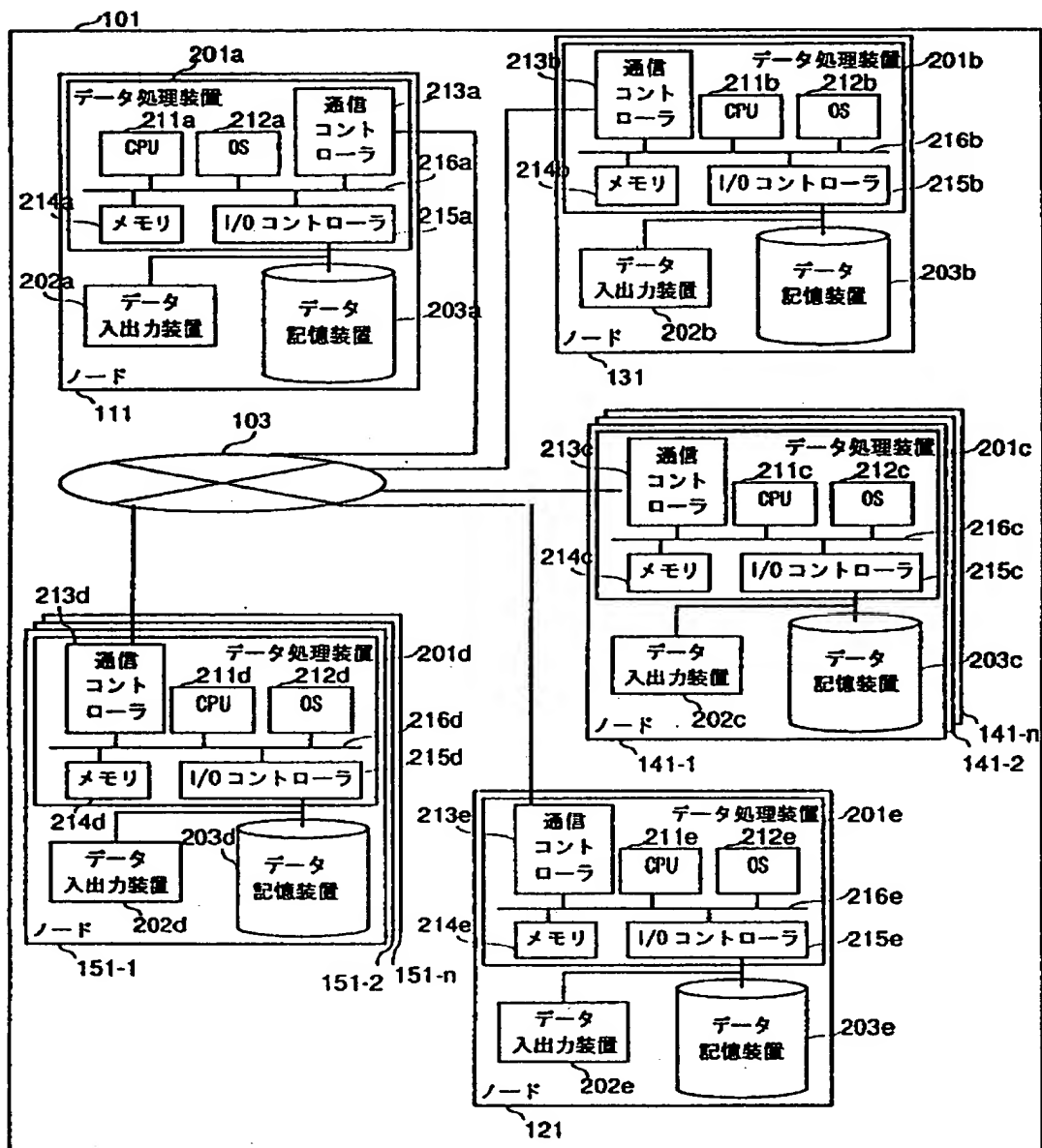
【図 1】

図 1



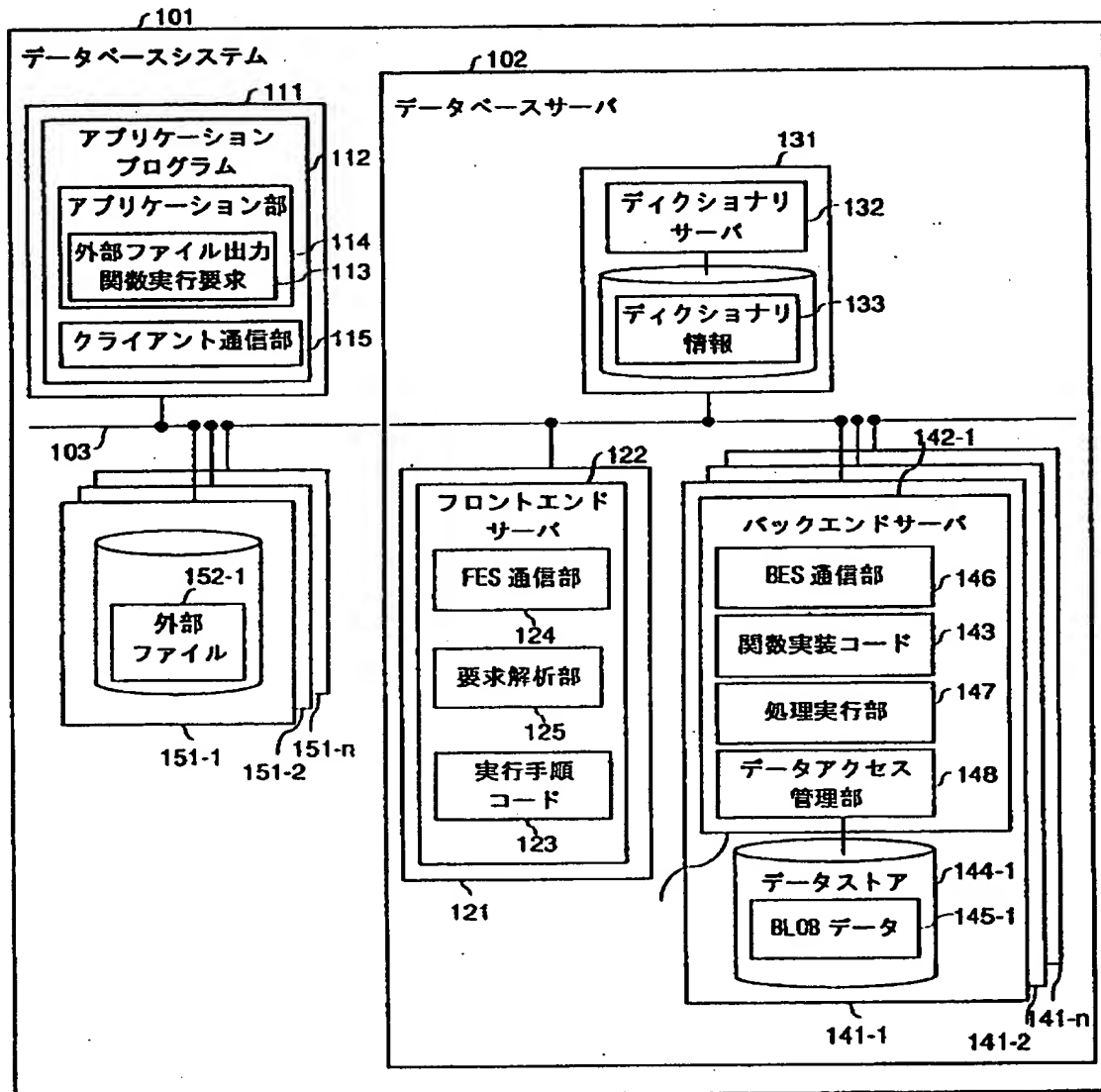
【図 2】

图 2



【図 3】

図 3



【図 4】

図 4

401
 CREATE TABLE employee (
 empno INTEGER, 402
 name VARCHAR(30), 403
 dept VARCHAR(30), 404
 photo BLOB(10M)); 405
 406

【図 5】

図 5

501 empno	511 name	512 dept	513 photo	514
1789	George	design	(blob1)	521
1797	John	account	(blob2)	522
1801	Thomas	design	(blob3)	523
1809	James	general	(blob4)	524
1829	Andrew	account	(blob5)	525
1837	Martin	planning	(blob6)	526

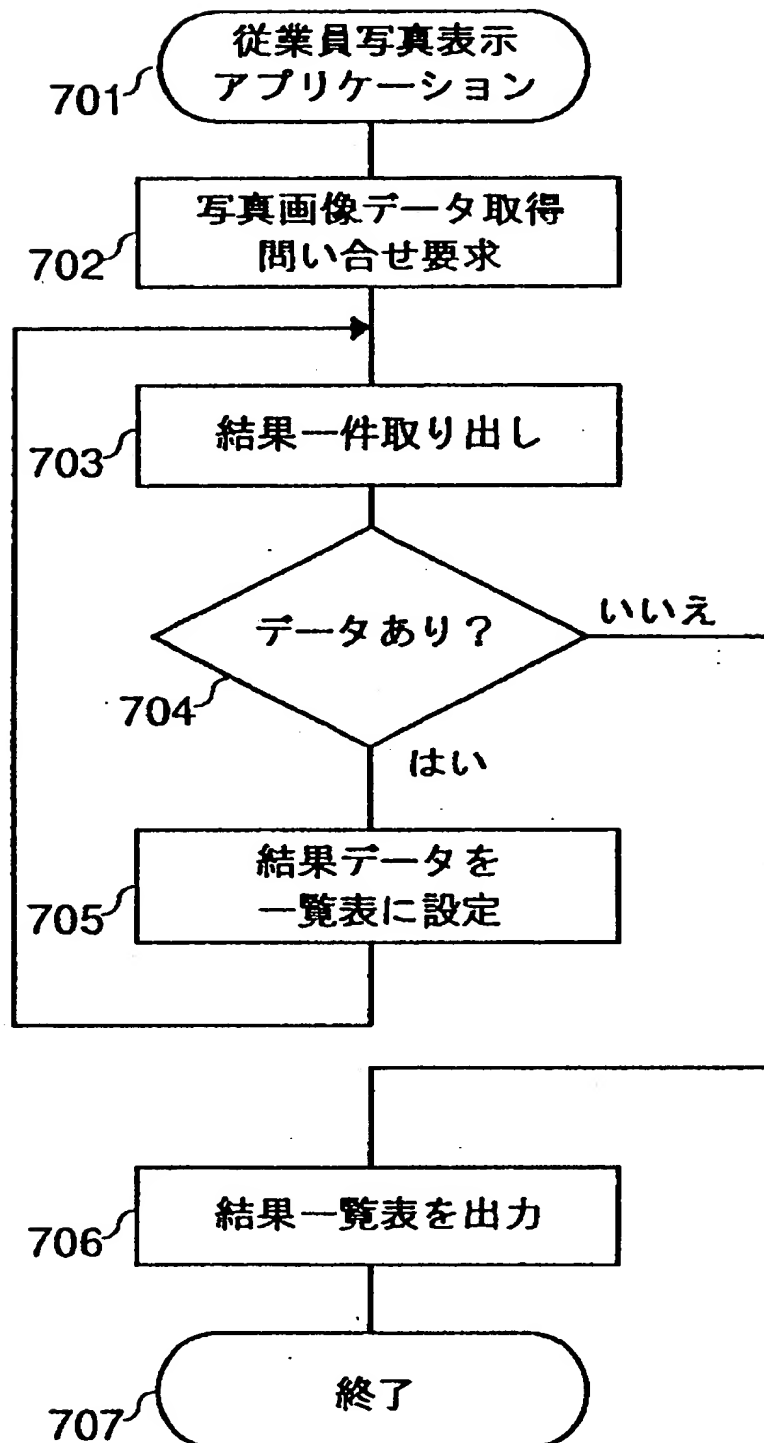
【図6】

図 6

601
CREATE FUNCTION fileout (BLOB) 602
RETURNS VARCHAR(255) 603
EXTERNAL NAME 'bin/fileout' 604
LANGUAGE C ; 605

【図 7】

図 7



【図8】

図8

```

801
EXEC SQL BEGIN DECLARE SECTION      802
  VARCHAR(255) photoFilename;      803
  VARCHAR(30) empName;              804
EXEC SQL END DECLARE SECTION;      805

EXEC SQL DECLARE curs1 CURSOR FOR 806
  SELECT name, fileout(photo)      807
  FROM employee                     808
  WHERE dept = 'design';            809

EXEC SQL OPEN curs1;                810

while (1) {                          811
  EXEC SQL FETCH curs1 INTO :empName, :photoFilename; 812

  if (SQLCODE == 100) break; /* no more data */ 813

  setImageDataToList(empName, photoFilename); 814
}                                     815

EXEC SQL CLOSE curs1;               816

displayImageDataList();              817
  
```

【図9】

図9

901	911	912	
	George	dbsvexfile1	921
	Thomas	dbsvexfile2	922

【図 10】

図 10

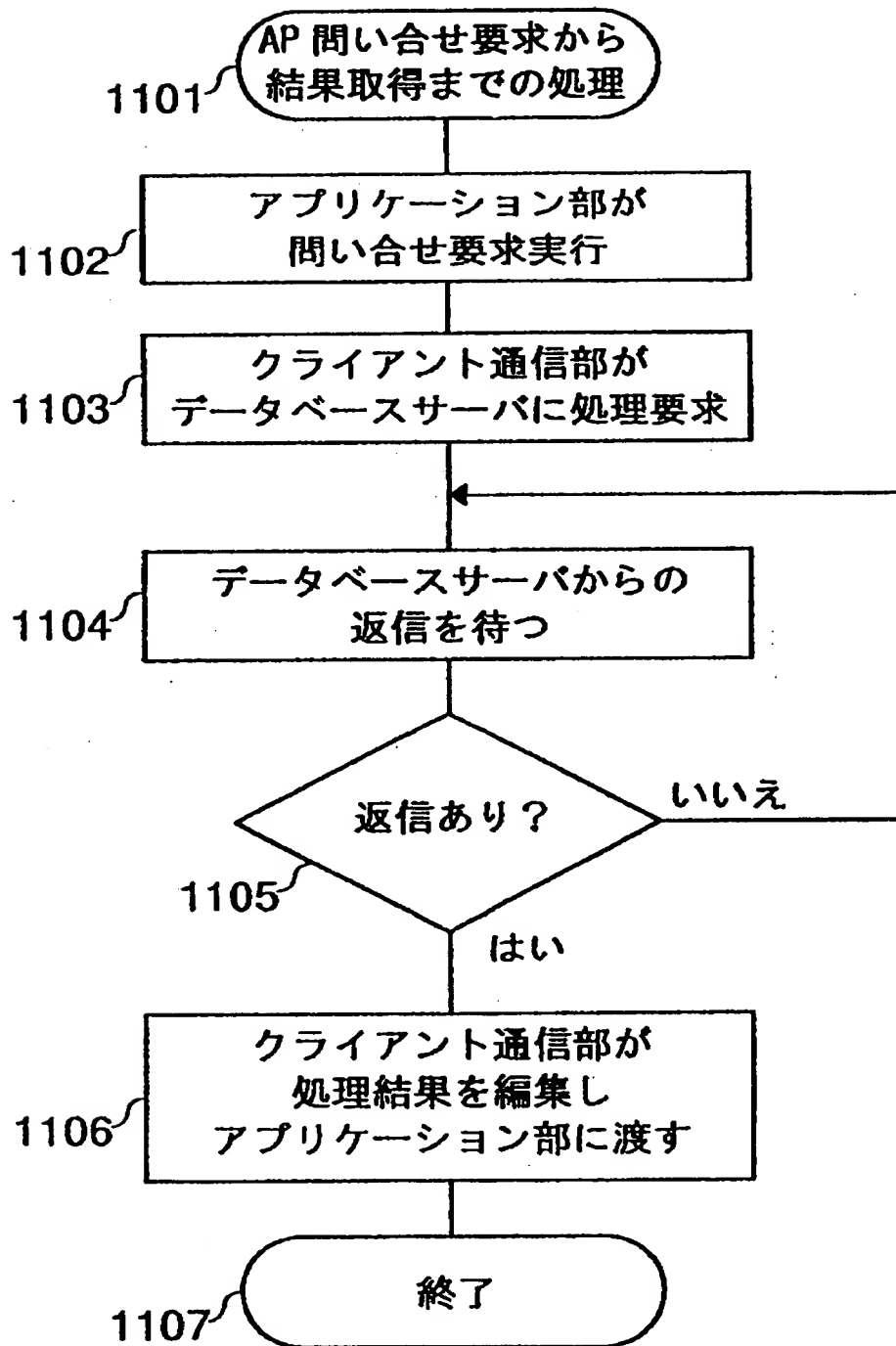
1001

```
<HTML>
  George: <IMG SRC="dbsvexfile1"><BR>
  Thomas: <IMG SRC="dbsvexfile2"><BR>
</HTML>
```

1002
1003
1004
1005

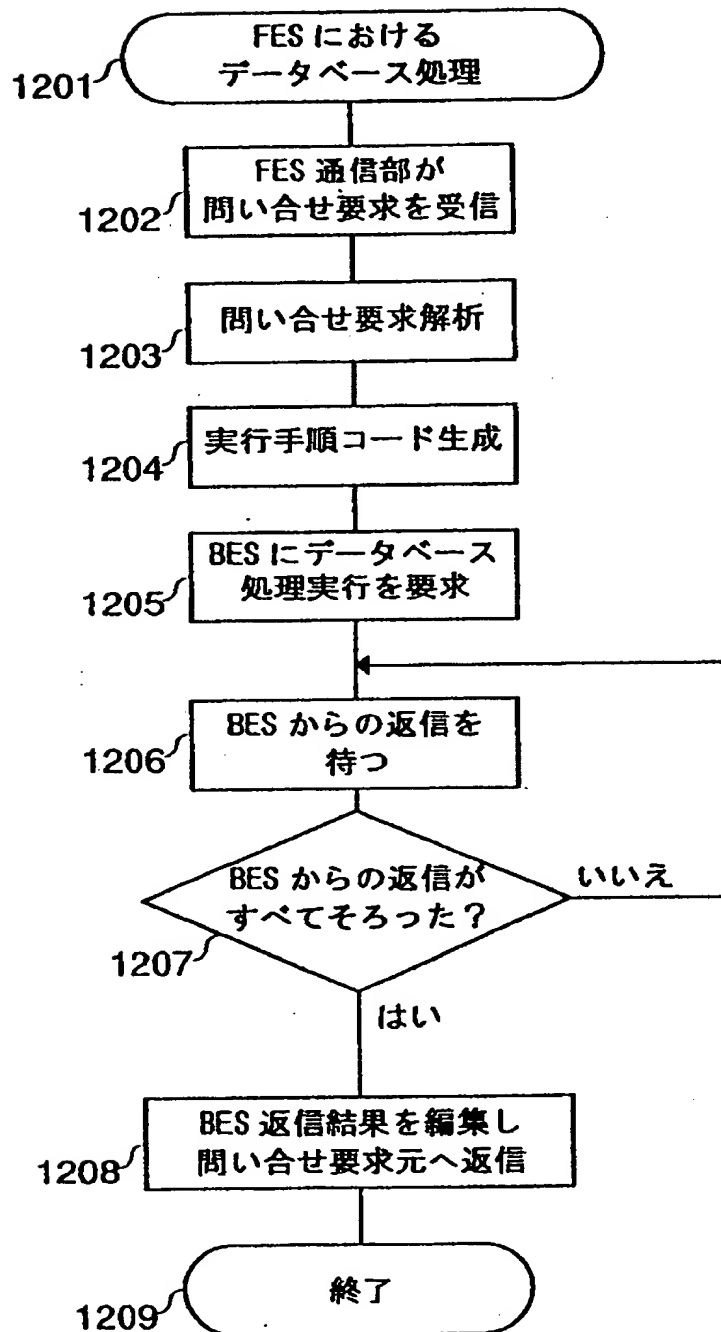
【図 11】

図 11



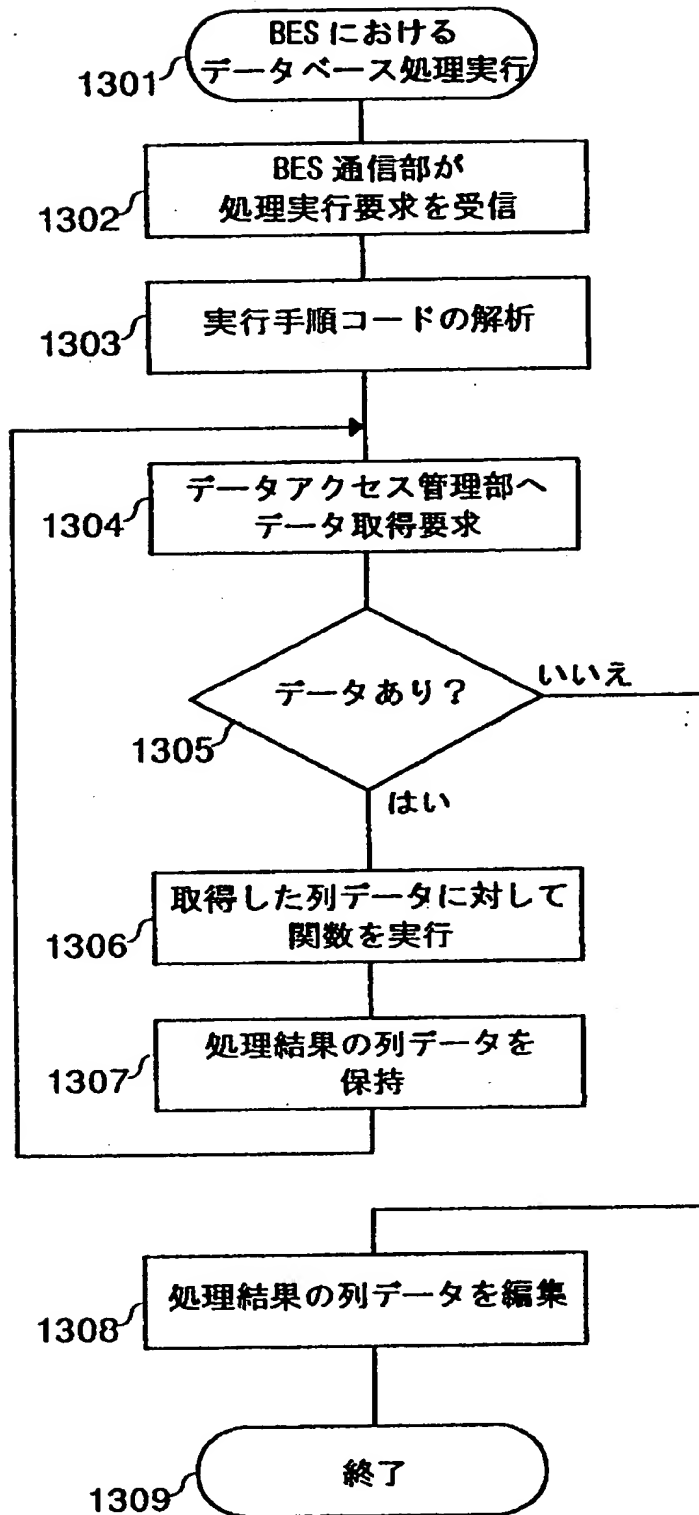
【図 12】

図 12



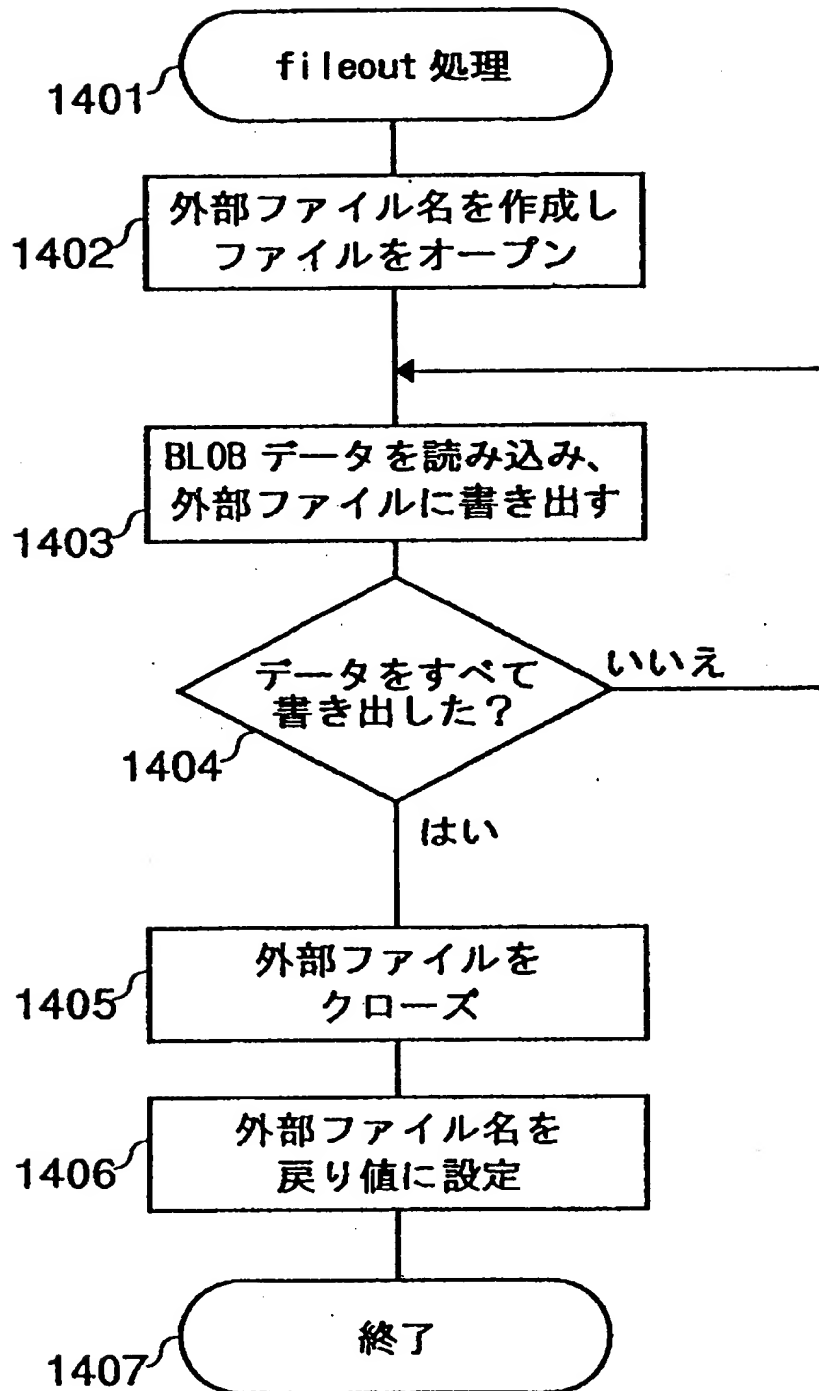
【図 13】

図 13



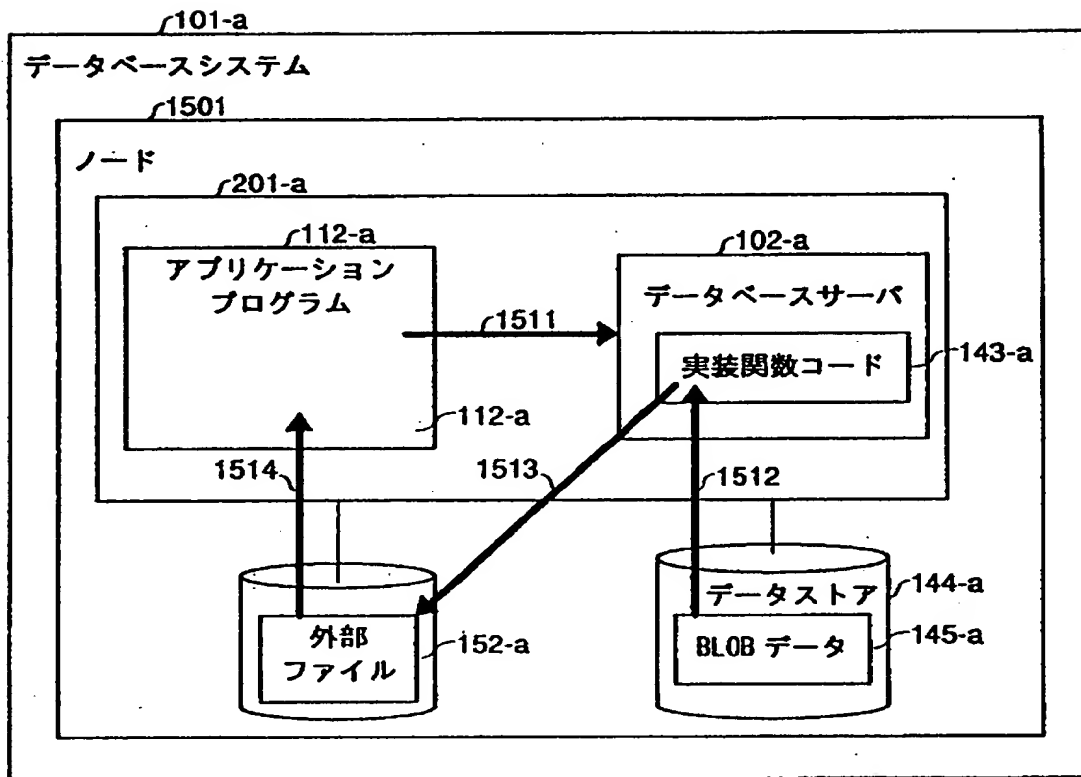
【図 14】

図 14



【図 15】

図 15



【書類名】 要約書

【要約】

【課題】 クライアントーサーバ構成のデータベースシステムにおいて、外部ファイルを用いて長大データを扱う処理を高速に行なうことを課題とする。

【解決手段】 クライアントーサーバ構成のデータベースシステムにおいて、サーバで動作するデータベースサーバがクライアントで動作するユーザアプリケーションから要求されたデータベースに格納されている長大データをファイルに出力する手段と、前記手段により長大データが出力されたファイルを前記ユーザアプリケーションが参照して長大データを取得する手段からなる。

【選択図】 図1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000005108

【住所又は居所】 東京都千代田区神田駿河台四丁目6番地

【氏名又は名称】 株式会社日立製作所

【代理人】 申請人

【識別番号】 100083552

【住所又は居所】 東京都荒川区西日暮里6丁目53番3号 藤井ビル
201号

【氏名又は名称】 秋田 収喜

特平10-147262

出 願 人 履 歴 情 報

識別番号

[000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地

氏 名 株式会社日立製作所